

Podstawy Programowania

Laboratorium 3

Funkcje i podstawowe algorytmy przetwarzania tablic

prowadzący: mgr inż. Marta Lampasiak, mgr inż. Michał Jaroszczuk

1 Wprowadzenie

Celem zajęć jest poznanie podstaw programowania w języku C/C++, w szczególności ćwiczenia z tworzeniem programów ilustrujących wykorzystanie reprezentacji tablicowej. Przetwarzanie tablic za pomocą pętli. Wybrane algorytmy przetwarzania tablic: wyszukiwanie liniowe i binarne, sortowanie bąbelkowe i przez wstawianie. Funkcje z jawną listą argumentów. Przekazywanie argumentów przez wartość, referencję i adres. Debugowanie i testowanie poprawności programów.

W celu zachowania porządku w kodzie programu warto fragmenty programu wydzielić w postaci funkcji. W szczególności jeśli do wydzielonego fragmentu mamy zamiar odwoływać się w różnych miejscach głównego programu. W przypadku funkcji zawsze należy pamiętać, że jej typ musi być zgodny ze zwracaną wartością. Do wnętrza funkcji można przekazać wartości w formie parametrów. Przy deklarowaniu parametrów (wewnątrz nawiasów okrągłych) należy pamiętać, że trzeba określić typ każdego parametru. W momencie wywołania funkcji domyślnie są przekazywane kopie podanych wartości. Jeśli chcemy mieć możliwość zmiany oryginalnej zmiennej to możemy zrobić to przy użyciu referencji, np.: `int ¶m`.

```

1 #include <iostream>
2
3 int length = 20;
4
5 int rnd(int down, int up, bool withup) {
6     return down + rand() % (up - down + withup);
7 }
8
9 int main()
10 {
11     srand(0);
12     for (int i = 0; i < length; i++ )
13         std::cout << rnd(1, 10, true) << " ";
14     std::cout << "\n";
15 }
```

Do parametrów oraz zmiennych lokalnych utworzonych wewnątrz funkcji mamy jedynie dostęp wewnątrz ciała funkcji. Na powyższym listingu pokazano sposób wylosowania kolejno 20 liczb przy użyciu funkcji `rand()`. Funkcja ta zwraca wartość z przedziału $[0, 2^{15} - 1]$. Następnie została obudowana drugą funkcją, która pozwala losować liczbę z zadanego przedziału $[a, b]$ lub $[a, b]$ w zależności od użycia opcji `withup`. Warto zwrócić uwagę, że przy każdym uruchomieniu programu zawsze otrzymamy ten sam ciąg liczb: 9 10 9 8 6 8 6 6 1 3 4 1 3 2 8 2 6 6 8 1. Powodem jest ustawienie źródła (`seed`) losowania na stałą wartość: `srand(0)`.

Podczas nauki programowania spotkamy się z problemem przechowywania większej liczby danych. Szczególnie problematyczne byłoby deklarowanie wielu lub nawet dziesiątek zmiennych w tym celu. Z pomocą przychodzą zmienne tablicowe. W formie takiej zmiennej można zapisać dowolnie dużo wartości. Aby zadeklarować tablicę, należy wskazać jej typ, nazwę oraz rozmiar w nawiasach kwadratowych, np.: `int tablica[10]`, aby utworzyć tablicę przechowującą maksymalnie 10 liczb całkowitych. W celu przypisania lub odwołania się do konkretnego elementu tablicy również używamy nawiasów kwadratowych. Należy pamiętać, że adresy tablicy są z zakresu od 0 do `rozmiar-1`.

```

1 #include <iostream>
2 #define N 5
3
4 int numbers3[N];
5
6 void print(int tab[]) {
7     for (int i = 0; i < N; i++)
8         std::cout << tab[i]<<" ";
9     std::cout << "\n";
10    return;
11 }
12
13 int main()
14 {
15     int numbers1[N], numbers2[N] = {1,2,3};
16     print(numbers1);
17     print(numbers2);
18     print(numbers3);
19 }

```

Jeśli tablicę zadeklarujemy w formie zmiennej globalnej to domyślnie będzie „wypełniona” zerami, w przeciwnym wypadku wartości mogą być różne w zależności od używanego środowiska. Podczas deklaracji tablicy można zainicjalizować wartości początkowe. W przypadku niepodania wszystkich wartości pozostałe zostaną uzupełnione zerami. Na powyższym przykładzie pokazano sposób przekazania tablicy do funkcji. Należy pamiętać, że w zmienne tablicowe są zawsze przekazywane w formie referencji. Efekt działania powyższego listingu, będzie następujący:

```

-858993460 -858993460 -858993460 -858993460 -858993460
1 2 3 0 0
0 0 0 0 0

```

Tablice muszą mieć zawsze określony stały rozmiar. Do zdefiniowania stałej możemy posłużyć się dyrektywą `#define`. Nie musi też zawsze korzystać z całej szerokości tablicy.

2 Zadanie

1. OCENA 3:

Napisz program do obsługi tablic jednowymiarowych przy użyciu funkcji. Rozmiar tablicy może być zakodowany na sztywno i nie mniejszy niż 5. Nie należy używać tablicy alokowanej globalnie lecz przekazywać tablicę jako argumenty funkcji. Tablica powinna przechowywać elementy całkowite. Należy zapewnić następujące funkcje:

- Wprowadzenie kolejnych wartości do tablicy przez użytkownika,
- Zwrócenie sumy wszystkich elementów w tablicy,
- Zwrócenie wartości maksymalnej znajdującej się w tablicy,
- Wyświetlenie wartości zapisanych w tablicy w formacie:

```

tab[0]=...
tab[1]=...
tab[2]=...

```

- Wypisanie liczb, które pojawiły się w tablicy przynajmniej 3 razy.

2. OCENA 4:

Rozszerz program z zadania 1 o następującą funkcję:

- Uzupełnienie tablicy liczbami losowymi z zakresu podanego przez użytkownika. Niech losowane liczby przy każdym uruchomieniu programu będą inne.
- Zamianę wprowadzonej liczby dziesiętnej z zakresu 0-30 na odpowiadającą jej liczbę binarną, której cyfry są zapisane jako kolejne elementy tablicy.

Zaprezentuj działanie.

3. OCENA 5:

Uzupełnij program o funkcję dającą możliwość sortowania elementów tablicy poprzez sortowanie bąbelkowe. Zaprezentuj uzyskany efekt.

W kodzie pisanych programów proszę umieszczać komentarze. Brak komentarzy uniemożliwi zdobycie maksymalnej ilości punktów za zadania!

Uwaga

Przypominam o przesłaniu programów na koniec zajęć, według wcześniej podanego wzoru. Proszę o niedołączanie plików innych niż o rozszerzeniu `*.cpp`.